I. Modules & Used Methods

Our program involves four main modules, all of which seek to manipulate functions in a way or another to best describe a given data set. The minimization of the sum of squared errors obtained by taking the difference of the predicted plot values and values from the original data set is the main scheme in which this is achieved

A. Curve Fitting of Linear Forms

The first module in our program is **Linearized_Regression()** and it deals with forms that are naturally linear or nonlinear forms that are precedingly transformed into linearized ones, such linear forms represent a straight line when graphed as shown in [Fig. 1,1] the approach to minimize the squared errors here comprises in solving the linear system that corresponds to zero partial derivatives for each of the constants, a general formula for the relevant matrix equation is shown in the appendix.

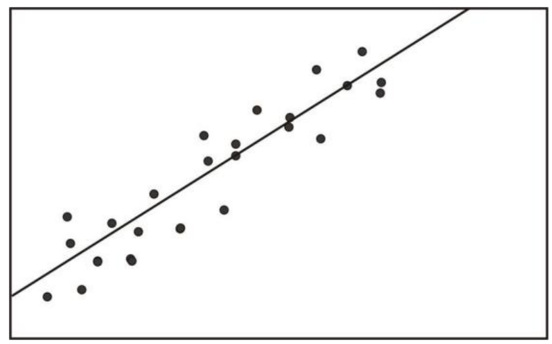

Fig. 1 Linear fitting of a data set

Inputs:

- The decimal fixing constant
- The data set $(x\ ,y)$
- The set of functions that correspond to the linearized form (limitless number)

Outputs:

- The linearized form with substituted constants
- The regression error

Additional Features:

- A plot involving the best fit and a scatter plot of the data set is shown through the **Plot_2D_RHS()** module

B. Curve Fitting of Nonlinear Forms

The second module in our program is **Nonlinear_Regression()** and it deals with forms regardless to whether they are linear or not, the approach to minimize the squared errors here comprises in the use of the Levenberg-Marquardt iterative scheme to find the minimum of the multivariable regression error function that involves all of the unknown constants, the Levenberg-Marquardt iterative scheme is an interpolation of the Gauss-Newton and Gradient Descent methods that are also used to find the minimum of multivariable functions, the Gauss-Newton algorithm has quadratic converges but is sensitive to the starting location, the Gradient Descent algorithm has linear convergence but isn't sensitive to the starting location, the Levenberg-Marquardt algorithm combines them to take advantage of both, [Fig. 2,2] illustrates the process for a form that involves two constants besides of the dependent variable, the iterative scheme of each of the algorithms is shown in the appendix.

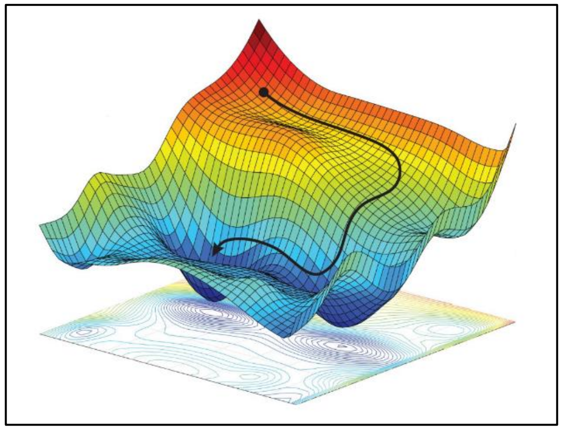

Fig. 2 Finding the minimum of a multivariable function.

Inputs:

- The decimal fixing constant
- The data set $(x\,, y)$
- The nonlinear function in terms of the dependent variable and the constants (up to three)

Outputs:

- The nonlinear function with the constants substituted
- The regression error

Additional Features:

- A plot involving the best fit and a scatter plot of the data set is shown through the **Nonlinear_Plot()** module

D. The Curve Family Detective

The fourth module in our program is **Curve_Family_Detective()** and what it does is receive a data set and then try to predict which family of curves best describes this data set, the approach here comprises in computing the best fit for several curves (linear, quadratic, cubic, power , exponential, logarithmic, reciprocal) and then selecting the family which correlates to the least regression error.

Inputs:

- The decimal fixing constant
- The data set (x, y)

Outputs:

- The family of curves that best describes the data set with its best fit
- Its regression error
- The standard deviation of the set of regression errors due to other family curves

---

II. Applications in Computer Engineering

Curve fitting has many applications, typically it's used to arrive at a stable model that can describe an ongoing system by just using discrete pieces of information about that system, this could be used to derive physical equations, predict the stock market or even better, serve as the basis for all of machine learning and artificial intelligence.

Curve Fitting & Artificial Intelligence

From beating world chess champions, repainting art pieces from Van Gogh and helping scientists fight the COVID-19 pandemic, AI has shown the world that the potential of technology might be boundless after all, but anyway, how could it ever be related to curve fitting? the way that curve fitting contributes in machine learning and artificial intelligence is implicit, they both work in the same ways, so much that whole books have been written on how this is true.

V. References

[1] *Least-Squares Regression*. https://www.ck12.org/statistics/least-squares-regression/. Accessed 14 May 2020.

[2] Shetye, Sweta. *Gradient Descent*. 2020, https://datavyom.com/2020/05/02/gradient-descent/. Accessed 14 May 2020.

[3] Munteanu, Marian. *The Paraboloid Of Revolution*. 2008, https://www.researchgate.net/figure/The-paraboloid-of-revolution_fig1_2207021. Accessed 14 May 2020.

VI. Appendix

The General Regression Formula:

For:

$$Y = C_0\emptyset_0 + C_1\emptyset_1 + C_2\emptyset_2 + C_3\emptyset_3 + \cdots + C_n\emptyset_n$$

Solve the matrix equation:

$$\begin{pmatrix} \sum \emptyset_0\emptyset_0 & \sum \emptyset_0\emptyset_1 & \sum \emptyset_0\emptyset_2 & \cdots & \sum \emptyset_0\emptyset_n \\ \sum \emptyset_1\emptyset_0 & \sum \emptyset_1\emptyset_1 & \sum \emptyset_1\emptyset_2 & \cdots & \sum \emptyset_1\emptyset_n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum \emptyset_n\emptyset_0 & \sum \emptyset_n\emptyset_1 & \sum \emptyset_n\emptyset_2 & \ldots & \sum \emptyset_n\emptyset_n \end{pmatrix} \begin{pmatrix} C_0 \\ C_1 \\ \vdots \\ C_n \end{pmatrix} = \begin{pmatrix} \sum \mathrm{Y}\emptyset_0 \\ \sum \mathrm{Y}\emptyset_1 \\ \vdots \\ \sum \mathrm{Y}\emptyset_n \end{pmatrix}$$

For surface fitting replace Y with Z and $\emptyset$ becomes a function in both x and y.

Gradient Descent Iterative Scheme:

$$x_{i+1} = x_i - \lambda\nabla f(x_i)$$

Gauss-Newton Iterative Scheme:

$$x_{i+1} = x_i - \left(\nabla^2 f(x_i)\right)^{-1}\nabla f(x_i)$$

Livenberg-Marquardt Iterative Scheme:

$$x_{i+1} = x_i - (H + \lambda I)^{-1}\nabla f(x_i)$$